

# CodeSqueezer Basic

## 概要

CodeSqueezer Basic は、四則演算を行う Verilog ソースコードを簡単な画面操作で作るソフトである。通常の IP が生成論理の性能を重視するのと異なり、このソフトはコーディング作業の生産性向上に主眼を置いている。このため、単純なボタン操作で最適な出力ポート定義を形成したり、他のソースコードに含まれる信号線定義を取り込む機能を備え、オーバーフロー発生時のフラグ出力やクリッピング処理を行う論理の形成、固定小数点位置などを自動処理している。出力は Verilog のソースコードであり、改変も可能である。現在、全機能を備えた評価版が無償公開されているが、評価版の生成コードを顧客製品に組み込みむことは許されていない。2 月中旬に発売される製品版は出力コードの顧客製品への組み込みが無制限に可能となり、知的財産権によって将来の製品展開を束縛されるリスクも回避されている。

## 1. 入手方法

CodeSqueezer Basic には評価版と製品版があり、シグナル・プロセス・ロジック社の HP<sup>1</sup>から入手することができる(製品版は 2010 年 2 月の発売予定となっている)。

いずれも Microsoft Windows Installer 形式で提供されており、画面の指示に従ってキー操作することで容易にインストールすることができる。必要に応じてマイクロソフト社のサイトより .NET Framework が自動インストールされる。

CodeSqueezer Basic を最初に起動した際には、ライセンス条件を確認する画面が表示される。“START”ボタンを押して使用条件を読み“I Agree”ボタンを押すことで全機能が使用可能となる。

## 2. 機能の選択

(select function)と書かれたボックス右側の“▼”マークをマウスクリックすると機能一覧が表示される。マウスを動かして必要な機能名を反転表示させた状態で、マウスを左クリックすることにより機能が選択される。

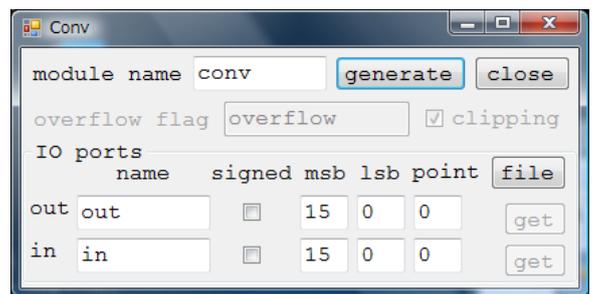
選択できる機能は以下のとおりである。

- ・ **Conv:** 信号表現形式の変換
- ・ **Shift:** シフト
- ・ **AddSub:** 加算と減算
- ・ **Multiply:** 乗算
- ・ **Divide:** 除算
- ・ **Encode:** エンコード
- ・ **Normalize:** 正規化

以下、個々の機能について解説する。

## 3. 信号表現形式の変換(Conv)

機能“Conv”を選択すると下図の Conv 機能ウィンドウがポップアップする。この機能は入力(in)と出力(out)の信号線表現形式を変換するもので、信号線表現形式として符号の有無、ビット幅、小数点位置が指定可能である。



Conv 機能ウィンドウ

これらの信号線表現形式を与えた結果、出力信号がオーバーフローする可能性がある場合は、オーバーフロー信号名を指定するボックスと、クリッピング処理の有無を指定するチェックボックスが操作可能となる。

オーバーフローの可能性があり、かつその信号名が指定された場合、形成されるモジュールにはオーバーフロー信号を出力するポートが追加される。

クリッピング処理が指定された場合、オーバーフロー発生の際の出力信号は、表現可能な最大値または最小値に固定される。

## 4. ポート定義の自動取り込み

CodeSqueezer Basic には既に定義済みの信号線定義を取り込む機能がある。

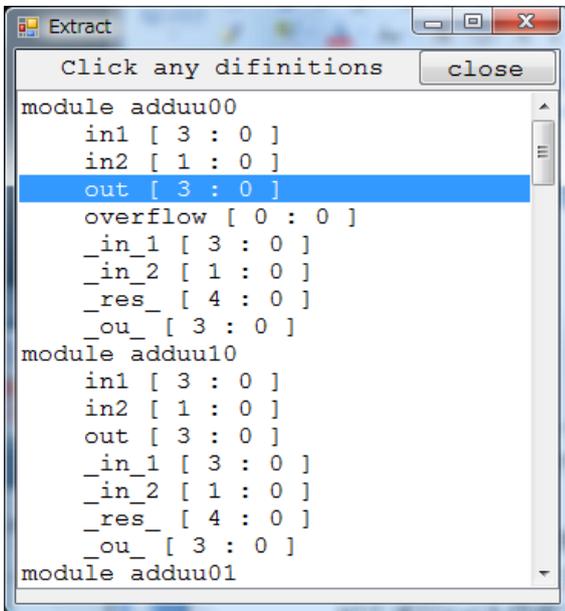
“file”ボタンをクリックすると[開く]ダイアログウィンドウが開くので、利用したい信号線定義を含む VerilogHDL ソースファイル(通常テキストファイルでもかまわない)を指定

<sup>1</sup> <http://signal-process-logic.com>

する。

CodeSqueezer Basic は、指定したファイルに含まれる信号線定義部分を自動的に検出し、“Extract”ウィンドウに表示する。

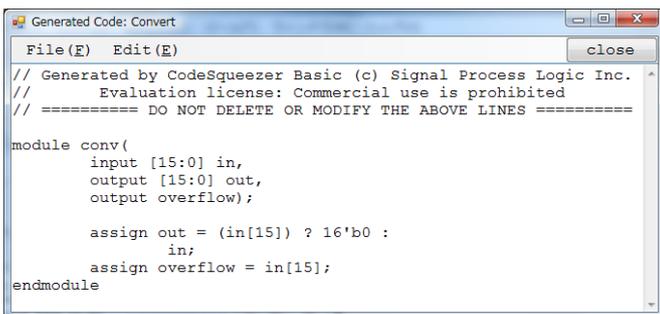
“Extract”ウィンドウに表示された信号線定義行をマウス左クリックで反転表示させ、機能ウィンドウ内の“get”ボタンを押すと、反転表示された信号線定義が“get”ボタンに対応するポート定義部分に取り込まれる。



Extract ウィンドウ

## 5. モジュール形成と結果の出力

機能ウィンドウの“generate”ボタンを押すと、指定された条件に合わせて、モジュールが自動形成される。



Generated Code ウィンドウ

上の図は、自動生成された 16 bit の符号付き整数を同じビット幅の符号なし整数に変換するモジュールであり、入力信号が負の場合にオーバーフローが発生して出力信号を最小値 0 にクリッピングするコードが形成されている。

このウィンドウに表示されたモジュールは、File プルダウンメニューによりファイルセーブすることができるほか、Edit プルダウンメニュー、マウス操作あるいはショートカットキー

(Ctrl-A, Ctrl-C)によりクリップボードにコピーし、ユーザーのソースコードにペーストすることができる。

なお、このウィンドウに表示されたソースコードはユーザーが任意に編集可能であるが、先頭に記述された3行のコメントはこのモジュールの著作権表示を行う部分であり、変更したり削除したりしてはならない。

## 6. シフト(Shift)

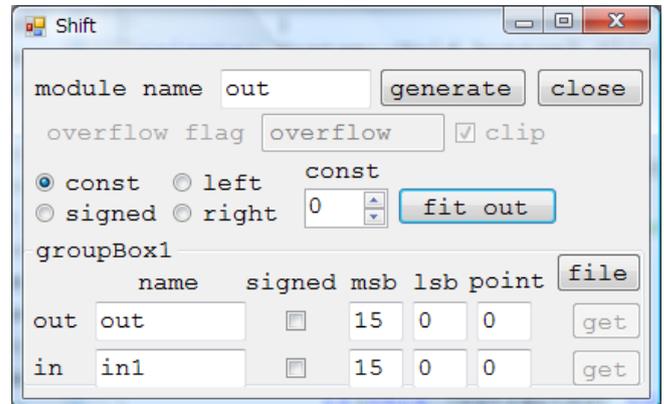
次図はシフト演算モジュールを形成するための機能ウィンドウであり、次の4種類がある。

**定数シフト:**一定数だけ左右にシフトする

**符号付きシフト:**与えられた数だけ左右にシフトする

**左シフト:**符号なしで与えられた数だけ左にシフトする

**右シフト:**符号なしで与えられた数だけ右にシフトする

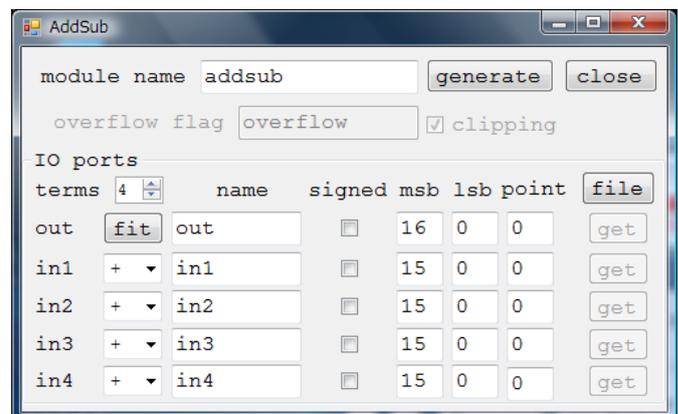


Shift 機能ウィンドウ

定数シフトおよび符号付シフトに際しては、シフト数が負で与えられた場合には右シフトとなる。

## 7. 加算と減算(AddSub)

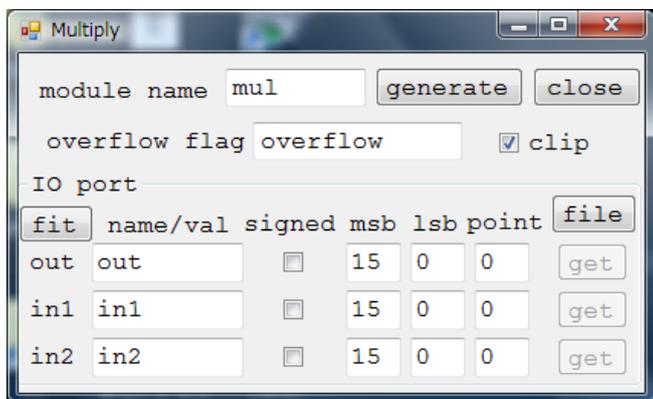
加減算モジュール形成機能は、2~4項の加減算を一つのウィンドウで形成することができる。このために使用される AddSub 機能ウィンドウを下に示す。



AddSub 機能ウィンドウ

## 8. 乗算(Multiply)

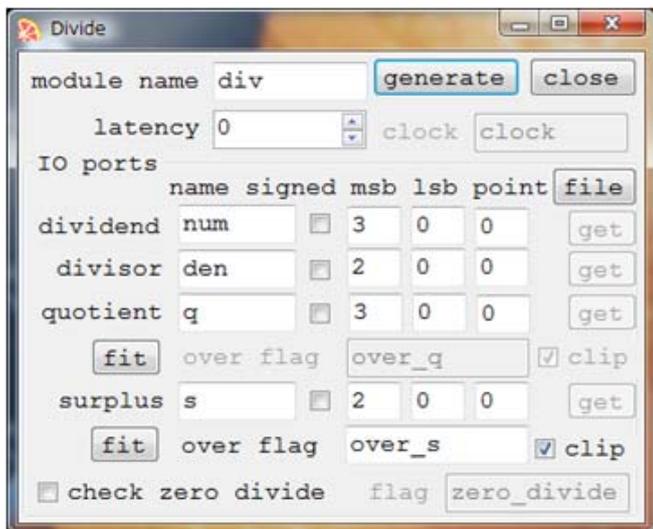
CodeSqueezer Basic は論理構成による乗算モジュールを生成することができる。乗算機能ウィンドウを以下に示す。ただし、この機能で形成される乗算論理よりは、FPGA に備わっている乗算機能ブロックを用いたほうが効率的である場合が多い。



Multiply 機能ウィンドウ

## 9. 除算(Divide)

CodeSqueezer Basic は、符号付引き放し法に基づく除算モジュールを自動形成する機能を持つ。除算モジュール形成のための機能ウィンドウを下に示す。



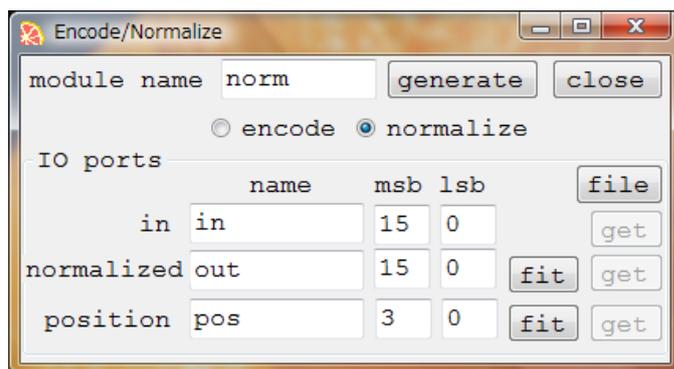
Divide 機能ウィンドウ

除算は除数の桁数に応じて論理段数が増大するため、パイプライン化も可能としており、latency 部に指定した段数のレジスタが論理の間に挿入される。

負数の除算には、商を 0 に近づける形に絶対値を切り捨てる手法と、剰余を常に正とする手法の二通りある。前者は C 言語など採用している手法であるが、符号が変化する前後で商の変化が一様でないという問題がある。本ソフトウェアは剰余を常に正とする手法を採用してこの問題を回避している。

## 10. エンコード(Encode)と正規化(Normalize)

エンコードモジュールと正規化モジュールの形成には数に示す同じ機能ウィンドウを用いている。



Encode/Normalize 機能ウィンドウ

エンコードとノーマライズは、いずれも入力信号線を最上位が 1 となるまで左シフトし、この際のシフト数を出力する。また、最上位が 1 となるまでシフトを行った信号は正規化された信号となる。

## 11. まとめ

CodeSqueezer Basic が形成する数値演算に採用されているアルゴリズムは、いずれもありふれたアルゴリズムである。しかしながら、教科書に掲載されているようなアルゴリズムであっても、これを理解し、用途に合う形に信号線を定義して HDL で記述する作業にはかなりの時間を要する。

CodeSqueezer シリーズは、このような作業時間を短縮してコーディング作業の生産性向上を実現するツールであり、あらかじめ用意された各種四則演算コードの雛形に適宜変更を施したものを出力する。

CodeSqueezer Basic には、コーディング作業を支援する種々の機能を備えており、極めて短時間のうちに所望の機能をもつ四則演算モジュールをソースコードの形で得ることができる。

HDL コーディング作業の生産性を改善するツールには、プログラマを煩雑な業務から解放するとともに、関連企業の収益を改善することが期待される。

シグナル・プロセス・ロジック株式会社は、今回発表の固定小数点四則演算を対象とする CodeSqueezer Basic を皮切りに、各種の汎用的な論理モジュールを簡単操作で形成するソフトウェアとしてシリーズ化を計画している。今後の製品群にも御期待いただきたい。

## Appendix: CodeSqueezer Basic ソフトウェアの使用条件

CodeSqueezer Basic は、最初に起動した際に使用条件が表示され、「I Agree」ボタンを押してこれに同意しない限りソフトウェアを使用することはできない。

使用条件には評価版と商用版の二通りあり、それぞれの版の使用条件は、概ね以下のとおりである。

### A1. 評価版の使用条件

1. 無保証:本ソフトウェアを使用することで生じるあらゆる結果に対して弊社(シグナル・プロセス・ロジック株式会社)は責任を持ちません。
2. 再配布の禁止:本ソフトウェア(複製物を含む)を再配布することはできません。他者に本ソフトウェアの使用を勧める場合は弊社ホームページの URL をご紹介ください。
3. 著作権の帰属:本ソフトウェアによって生成されたコード(以下「生成コード」という)の著作権は弊社に帰属します。
4. 生成コードの変更:生成コードは、先頭のコメント部分を除き、使用者が任意に変更することができます。変更部分の二次著作権は使用者に帰属しますが、一次著作権は引き続き弊社に帰属します。
5. 生成コード利用の許諾:弊社は、次のいずれかの場合に限り生成コードの利用を許諾します。
  - 5.1 本ソフトウェアの機能を評価する目的でなされる生成コードの複製
  - 5.2 生成コードを利用した製品に関わるソースコードを含む技術情報が公表され、かつ該製品およびこれに関わる技術情報の複製・頒布が無許諾・無償で認められる場合の、該製品の開発及び流通に関わる生成コードの複製および頒布なお、上記以外であっても私的利用や引用などの著作権法が定める著作権制限条項に該当する場合は、生成コードを無許諾・無償で利用することができます。

ここで、各用語の意味は以下のとおりです。

- 「本ソフトウェアを使用する」とは、CodeSqueezer Basic の各機能を実行させることを意味します。
- 「生成コードを利用する」とは、生成コードの複製物を作成したり、これを第三者に提示したり譲渡したりする行為を指します。生成コードにコンパイル等の機械的処理を施して得られたものも、生成コードの複製物と同等に扱われます。

### A2. 製品版の使用条件

1. 本ソフトウェアは弊社が本ソフトウェアとともに供給するプロテクトキーを装着した1台の PC のみで使用することができます。
2. 本ソフトウェアによって生成されたコード(以下「生成コード」という)の著作権は弊社に帰属します。
3. 弊社は、本ソフトウェアおよびプロテクトキーの正当な所有者が自らの製品に組み込んだ形で行う生成コードの複製・頒布を無条件に許諾します。
4. 生成コードは、先頭のコメント部分を除き、使用者が任意に変更することができます。変更部分の二次著作権は使用者に帰属しますが、一次著作権は引き続き弊社に帰属します。
5. 弊社は本ソフトウェアが正常に作動するよう細心の注意を払っておりますが、生成コードの完全性を保証するものではありません。生成コードのご利用に際してはお客様の側で十分な検証を行われるようお願いいたします。
6. 本製品に関するご質問は電子メールで承ります。また、トラブル情報等につきましては弊社ホームページにて随時公開いたします。サポートに使用する言語は、当分の間、日本語に限らせていただきます。
7. お客様からお寄せいただきましたトラブル情報等は断りなく製品改良に使用場合があります。

制作・販売: **シグナル・プロセス・ロジック株式会社**

URL: <http://signal-process-logic.com>

お問合せ先: [cs@signal-process-logic.com](mailto:cs@signal-process-logic.com)